Secure and Efficient Digital Signature Scheme for Document Authentication using EDDSA and Watermarking

Sahibzada Hasanain[1], Qazi Ejaz Ali[1*], Abdul Haseeb Malik[1], Waheed Ur Rehman[1], Muhammad Haseeb[1], Saif Ur Rehman[1], Syed M. Asim Jalal[1], Tabinda Salam[2],
[1]Department of Computer Science, University of Peshawar 25120, Pakistan.
[2]Department of Computer Science, SBBWU, Peshawar, Pakistan.
*Correspondence: qaziejazali@uop.edu.pk

Citation | Hasanain. S, Ali. Q. E, Malik. A. H, Rehman. W. U, Haseeb. M, Rehman. S. U, Jalal. S. M. A, Salam. T, "Secure and Efficient Digital Signature Scheme for Document Authentication using EDDSA and Watermarking", IJIST, Vol. 8 Issue. 5 pp 1788-1813, August 2026

DOI | https://doi.org/10.33411/IJIST/1947

Received | July 09, 2026 **Revised |** July 31, 2026 **Accepted |** Aug 02, 2026 **Published |** Aug 13, 2026.

The growing use of electronic documents in the private business, legal and financial institutions has increased the demands of secure digital authentication systems which can provide strong security guarantees as well as sound forensic validation. However, many existing digital document authentication methods suffer from high computational and communication overhead, lack confidentiality measures and effective tampering detection techniques and also limited resistance to security attacks, particularly in multi-signatory environments. This paper introduces an efficient integrated digital signature scheme that combines Edward's curve Digital Signature Algorithm (EDDSA) for deterministic and nonce secure digital signing. Experimental results show that the proposed scheme has low signing time and high watermark correlation accuracy for tamper detection, and reliable identification of replayed and altered documents. The proposed scheme has a significant reduction in communication and storage overheads, significantly faster verification under large signatory groups, and improved resistance to sniffing and replay attacks.

Keywords: Digital Document Authentication; EdDSA (Ed25519); Multi-signature Scheme; AES-GCM Encryption; DWT Watermarking; Tamper Detection; Replay Attack Resistance; Secure Document Transmission; Cryptographic Integrity; Forensic Validation.

Introduction:

The rapid computerization of government, financial, and enterprise systems has increased reliance on electronic documents for communication, authorization, and record management. Distributed and cloud systems are now widely used to transmit contracts, financial statements, academic credentials, and regulatory filings [1]. Although this digitization enhances efficiency and accessibility, it also presents sensitive documents to security risks such as interception, manipulation, copying and illegal redistribution. Authenticity, integrity, confidentiality, and non-repudiation are essential for securing digital ecosystems and enabling reliable forensic validation [2]. The expansion of e-governance and electronic record systems increases the need for effective document authentication that have the ability to safeguard information against cyber-attacks and electronic forgery [3]. Digital forensic preparedness has also emerged as a prerequisite of the contemporary document authentication systems, especially in the context of the growing utilization of small-scale digital devices in organizational processes, including smartphones, internet-of-things devices and handheld storage devices. Forensic frameworks must ensure traceability, preserve evidence integrity, and detect tampering for legal and investigative processes within distributed computing environments [4]. These capabilities enhance the quality of evidence of digitally signed and encrypted documents in forensic studies.

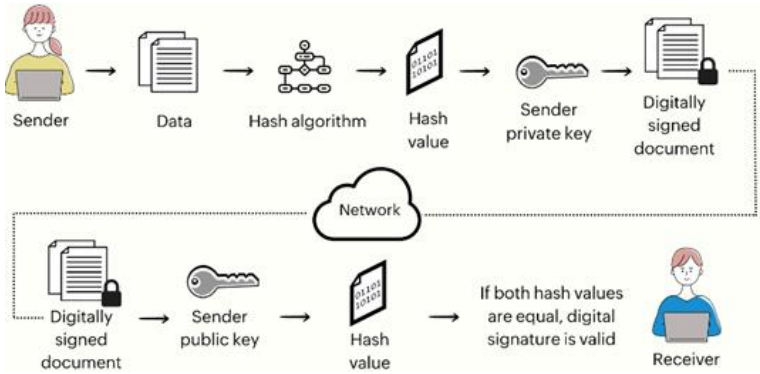

Figure 1. End-to-End Workflow of the Digital Signature and Authentication Scheme

Historically, the most popular method of authenticating electronic documents, as well as the identity of the signatories, has been the use of digital signatures as depicted in Figure 1. The initial public key signature systems, like RSA and Digital Signature Algorithm (DSA), laid the groundwork of the cryptographic authentication and non-repudiation of networked systems [5]. Later developments in elliptic curve cryptography led to the Elliptic Curve Digital Signature Algorithm (ECDSA) that offers similar security using smaller key sizes and better performance [6]. Such cryptographic methods find great applications in secure communications, digital contracts, and online verification systems [7]. Nevertheless, most of the current digital signature schemes have practical limitations, even though they have strong theoretical security assurances, such as excessive computational and communication costs, absence of confidentiality, and inadequate tamper detection. Most implementations do not encrypt signed documents, allowing interception and unauthorized access and man in the middle (MIM) attacks can affect the network transmission. In addition, nonce dependent signature algorithms can be attacked by weak or poorly designed randomness generation mechanisms that can lead to key exposure, replay attacks, and forgery of signatures [8]. These limitations require efficient authentication mechanisms with lower computational cost and stronger resistance to tampering, sniffing, and replay attacks in contemporary digital document systems.

Authenticated encryption addresses confidentiality issues during document transmission. A commonly used approach is the Advanced Encryption Standard in

Galois/Counter Mode (AES-GCM), which allows both data encryption and integrity verification to be performed in one cryptographic operation. AES-GCM is now commonly suggested as an authenticated encryption system due to its high tampering and sniffing attacks resistance, and relatively low computation cost [9]. These types of encryption are used to ensure that the contents of documents are not revealed to unauthorized persons when being transmitted, but encryption in itself does not help to identify who signed the document or to give legally binding evidence of authorship. As a result, encryption methods should be used in combination with digital signatures in order to provide extensive security of documents [10].

Digital watermarking complements cryptographic signing and encryption to identifying content manipulation and ownership of digital documents as it is presented in Figure 2. Watermarking incorporates a secret message like author identity, time-stamps or verification code into the structure of the documents, allowing forensic detection of forbidden changes when adversaries attempt to conceal their alterations [11]. Frequency domain watermarking techniques, especially the Discrete Wavelet Transform (DWT), offer high resistance to compression, scaling and format conversion with no loss to document quality [12]. Blind watermarking techniques are particularly useful since verification is possible without the knowledge of the original unwatermarked text and can be used in a distributed verification system [13]. Combination schemes that merge watermarking and cryptographic signature have also been investigated to enhance document safeguarding schemes within the digital setting [14]. However, watermarking is not sufficient to ensure cryptographic integrity or thwart replay attacks unless it is used with secure signing and encryption systems.

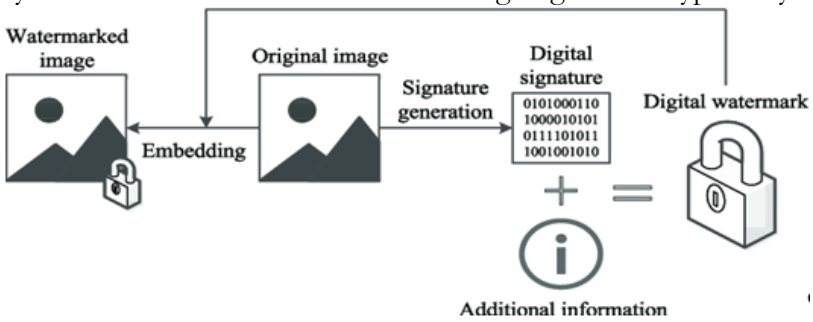

Figure 2. Watermarking Technique

New directions in elliptic curve cryptography have led to the Edwards curve Digital Signature Algorithm (EdDSA) as a deterministic and efficient variant of previous signature designs. To reduce weaknesses related to weak randomness, EdDSA derives nonce values in a deterministic manner, thus enhancing protection against key leakage and side-channel attacks [15]. Moreover, the current multisignature protocols can be used to synthesize a single aggregated signature by a group of multiple entities in order to minimize the number of communications and enhance the speed of verification when it comes to distributed approval [16][17]. These features are especially applicable in business and government processes where the documents usually need to be authorized by a number of stakeholders before they can be executed or archived [17]. The combination of elliptic curve-based signatures and watermarking methods to improve the overall security of electronic documents has also been studied recently [18][15]. Nonetheless, most of the current solutions continue to be based on partial security designs, do not include built-in confidentiality measures, or cannot support replay resistance and scalable multi-signature verification in cooperative settings [19]. Consequently, the solutions available tend to offer fragmented protection as opposed to a cohesive architecture that can support all the necessary security needs at the same time.

The paper provides a systematic methodological hierarchy whereby the first stage is to look at the development of digital signature technologies with specific focus on elliptic curve based deterministic and multi-signature algorithms. After that, the authenticated encryption

methods are considered to achieve secure document transmission and maintenance of confidentiality. The paper goes on to examine frequency domain watermarking methods, especially Discrete Wavelet Transform (DWT) based blind watermarking, to embed forensic metadata and allow detection of tampering. Lastly, these elements are combined into a single digital authentication system and tested experimentally in both a single signatory and multi signatory system to determine computational efficiency, communication overhead and resistance to replay and modification attacks.

The main aim of the paper is to develop and analyze a secure and efficient digital signature scheme that is used to ensure reliability in authenticating digital documents, detecting tampering, and protection against unauthorized access. The intended scheme will reduce the computational costs and enhance the appropriate metric of authenticated documents in distributed systems. Moreover, the scheme assesses its resistance to typical network attacks like sniffing and replay attacks. The novelty of this research is that it combines deterministic EdDSA based digital signature, AES-GCM authenticated encryption, blind DWT watermark embedding with a cryptographically secure pseudo-random number generator in a single verification architecture. This integrated method ensures authenticity, integrity, confidentiality, and non-repudiation while maintaining low communication overhead and strong resistance to document manipulation.

A few works have focused on digital signatures, watermarking techniques, authenticated encryption alone, or multi-signature authentication alone, but most hybrid authentication schemes combine only part of these security methods. In most of the methods reported, the integrity of the documents was focused, generally focusing on the use of digital signatures or watermarking, but with no attempt made to guarantee confidentiality or replay protection, no scalability for multi-signatures and no attempt to provide any sort of traceability. Most of the methods reported are focusing on document integrity, most commonly using digital signatures or watermarking, but with no aim of guaranteeing confidentiality and replay protection, with no scalability for multi-signatures and no attempt to provide traceability within a unified architecture.

The proposed framework provides a unified authentication (hybrid architecture) with multi-signature key aggregation using XOR, digital signatures using Ed25519, authenticated encryption using AES-GCM, a blind DWT watermark, and replay protection using timestamps in a single verification workflow. The proposed aggregation mechanism allows for the collaborative authorization of documents, with constant-sized authentication data, when the number of signatories grows, as in conventional multi-signature messages. Moreover, an embedding of a blind DWT watermark along with cryptographic authentication gives another forensic level in addition to the cryptographic validation, which acts as a means for "triangulation" and post-validating the documents to identify any tampering after cryptographic validation. The entire Tobacco3482 document set (6,964 document images) was used to implement the proposed framework and for its evaluation. Evaluation of experimental results covers authentication accuracy, security analysis, robustness under various image processing attacks, scalability for up to 2,000 signatories, comparison with the already available authentication schemes and computational complexity analysis of the algorithm, statistical performance analysis. These dual services enable the presented solution to outperform compared to any existing hybrid solutions.

Main Contributions:

The main contributions of this work are summarized as follows:

A unified hybrid authentication framework integrating Ed25519 multi-signature authentication, AES-GCM authenticated encryption, and blind DWT watermarking for secure document authentication.

An XOR-based multi-signature key aggregation mechanism that enables scalable authentication while maintaining constant communication overhead.

A replay-resistant verification protocol using timestamp validation integrated with authenticated encryption.

Extensive experimental evaluation on the Tobacco3482 benchmark containing 6964 document images, including watermark robustness against JPEG compression, Gaussian noise, cropping, scaling, and rotation attacks.

Scalability evaluation up to 2000 signatories together with computational complexity analysis, statistical confidence intervals, and comparison against existing authentication schemes.

A low-computation multi-signatory lightweight document authentication system using a multi-signature key aggregation mechanism based on XOR is proposed to provide scalable multi-signatory document authentication without high computation overhead.

Literature Review:

The issue of secure digital document authentication has gained great prominence with the use of electronic documents because of the high rate at which organizational workflows in government, health care, finance and education sectors are being digitized and the electronic documents need to ensure authenticity, integrity, confidentiality and non-repudiation. With the continued adoption of digital records to substitute paper-based records, the need to prevent forgery, tampering and unauthorized access of digital records is becoming a paramount necessity. As a result, scholars have examined some security tools such as cryptographic digital signature, encryption algorithms, digital watermarking, and multi-signature authentication systems to enhance the credibility of electronic documents and guarantee their admissibility in legal and forensic systems. Older methods are based on more popular cryptographic schemes like RSA, DSA, and elliptic curve-based signature, with newer literature exploring deterministic schemes like EdDSA and hybrid security designs, which combine encryption and watermarking together to enable greater security protection.

The cryptographic methods of document authentication are usually divided into symmetric encryption techniques, asymmetric public key cryptography techniques and hybrid security scheme. Symmetric encryption methods largely revolve around the provision of confidentiality by the use of a common secret key between the two parties. Other scholars have adopted message authentication which is a combination of HMAC and SHA-256 to authenticate the integrity of documents during transmission. Similar to, [20] suggested a document authentication method based on HMAC SHA256 to create verification tags of electronic documents. Even though this technique offers protection against integrity, lack of public-key cryptography denies the possibility of independent validation of ownership of documents and removal of authentic non-repudiation. On the same note, [21] created a file authentication system based on HMAC-SHA256 to identify file manipulation before and after it is sent over the network. Although the system has ensured integrity of the messages, it is still susceptible to shared key compromise since any party with the key can produce valid authentication codes.

Symmetric encryption algorithms like AES, DES, Blowfish, RC4, and ChaCha20 have been tested in other studies to secure information being transferred over a network. [22] compared these algorithms based on their performance and confidentiality protection of secure database communication. Although symmetric encryption is effective in maintaining secrecy of data, it lacks the ability to provide authentication of the source of the document or authenticate authorship, which exposes systems to MIM attacks and unauthorized alteration, when implemented without complementary authentication systems. On the same note, [23] suggested a watermark dependent authentication system in which encrypted watermark patterns are coded into document structures in order to verify the document using forensic

authentication. Despite the fact that this method enhances tamper detection, lack of cryptographic binding between the signer and the document undermines the non-repudiation properties. Likewise, [24] presented lightweight message authentication protocol based on symmetric MAC generation on restricted devices, which have shown better performance in low-power systems but have proven to be vulnerable in situations where shared keys are revealed. [9] also applied symmetric HMAC based authentication to distributed systems which incurred less computational overhead as compared to asymmetric signatures but they observed challenges in distributing and managing keys in open network systems.

As a way to overcome the shortcomings of symmetric methods, most researchers have turned to asymmetric public key cryptography to authenticate a digital document. Public key systems run with mathematically related pairs of keys of a public key and a private key, thereby providing secure communication without shared secrets. Commonly applied algorithms are RSA, DSA and Elliptic Curve Cryptography (ECC). A number of researches have applied RSA based digital signatures to authenticate electronic documents. As an example, [25] have suggested a multifactor RSA digital signature scheme with signcryption to improve both confidentiality and authentication. Even though the method enhances security, it adds high computational cost and complexity of communication especially in cases where several signatories are targeted.

In the same manner, [26] came up with a selective digital signature algorithm in which RSA signatures are used in a selective region of structured documents to minimize the computation expenses. Although this technique is effective in enhancing processing speed of specific document types, its processing speed is lower when a document is filled with thick text or graphical data. Digital Signature Algorithm (DSA) implementations have been investigated in other studies as an authentication of a document. [27] combined DSA signatures and QR codes to authenticate academic documents and eliminate duplication of scanned signatures. Nonetheless, DSA uses very large key sizes and modular exponentiation operations, which are computationally intensive hence making signature generation and verification slower.

In an effort to enhance efficiency, researchers have shifted towards using elliptic curve-based signature algorithms like the Elliptic Curve Digital Signature Algorithm (ECDSA). To minimize the key size and computation cost of RSA and DSA, [28] suggested a QR code based digital signature system with ECDSA. Similarly, [17] proposed an optimized ECDSA authentication scheme to the resource constrained environment. Although these advances have been made, the implementation of ECDSA is still vulnerable to nonce generation, in which exposure of random numbers can reveal personal signing keys. In order to ensure these risks are mitigated, [18] came up with an improved ECDSA model that incorporates deterministic nonce generation that is compliant with RFC 6979 and watermark-based document verification to enhance provenance tracking.

More recently, researchers have explored the Edwards curve Digital Signature Algorithm (EdDSA) as an efficient and secure alternative to traditional elliptic curve signatures. [29] came up with a better version of EdDSA based document authentication system that was found to be much faster when executing in addition to having a high level of cryptographic security. Nevertheless, their structure does not support the functionality of multi-signature and lacks the elements of protection of confidentiality. A later EdDSA based multi-signature authentication system was proposed by [15] and was aimed at verifying sensitive identifiers (biometric identifiers and identification records). Although their system can have multiple people sign one document, jointly, it still does not have embedded encryption and is characterized by longer processing delays caused by the need to repeat the key generation process. Equally, [30] suggested a better EdDSA multi-signature scheme with the help of QR codes and cryptographically secure pseudo-random number generation to

create keys. Although these are the advances, lack of encryption leaves the transmitting documents to the risk of eavesdropping.

To cope with these issues, the hybrid cryptographic schemes have been suggested, which rely on asymmetric signatures, symmetric encryption, and digital watermarking methods. The objective of these hybrid techniques is to offer a multi-faceted guarantee of document protection, i.e. authentication, confidentiality, integrity, non-repudiation and tamper detection. [31] designed a layered security architecture that incorporated encryption as a confidentiality measure, digital signatures as an authentication measure, and watermarking as a measure of ownership. Even though the framework provides a strong level of protection to the documents, it is dependent on DSA, which presents high computational overhead. On the same note, [19] came up with a cloud-based digital signature system that incorporates DSA, Enhanced Data Encryption Standard (EDES) and watermarking to ensure security of electronic documents. However, the dependence of the system on third party cloud computing services introduces latency, computing load and complexity to the operations.

Although several studies combine cryptographic authentication with document protection mechanisms, existing approaches typically integrate only one or two security primitives, such as digital signatures with encryption or digital signatures with watermarking. Very few frameworks simultaneously address multi-signatory authentication, authenticated encryption, replay protection, and forensic watermarking within a unified architecture. These limitations motivate the proposed framework.

Material and Methods:

Investigation of Environment:

The study was conducted under a controlled computational research setup which was designed to emulate secure digital document exchange and authentication in distributed networks. As opposed to experimentally confined studies, which are geographically limited, the scope of this study is in a cyber operational environment where digital documents are created, processed, transmitted, intercepted, and validated under potentially adversarial conditions. The computational experiments were run on a workstation, with the processor being an Intel Core i5 and 8 GB RAM, Python 3.10. This medium size hardware configuration was to explicitly test the feasibility of the proposed structure in the context of a real-world institutional computing environment, not on the highly specialized high-performance infrastructure.

The experimental setup is a model of a multi signatory digital authentication process where documents are processed through phases of watermark embedding, cryptographic signing, encryption and verification and sent through untrusted communication channels as illustrated in Figure 3. Such a simulated environment is very appropriate in conducting research as it supports controlled testing of security properties such as confidentiality, authenticity, integrity and replay resistance. Moreover, it also allows conducting tampering experiments and performance benchmarking with reproducible conditions.

The suggested scheme is responsive to the new issues of digital document security whereby the conventional signature-based solutions are not effective in offering any forensic traceability or evidence of tampering. Combining cryptographic signature with transform domain watermark and using authenticated encryption, the investigation environment forms an appropriate testbed to assess robust document authentication mechanism that can be applied in electronic governance, in digital archiving and in secure institutional communication. Figure 3 depicts the sequence of major generation, signature creation, encryption, transmission, and verification operations, in the overall workflow of the proposed system.

Data Sources and Software Reliability: The experimental dataset consisted of the Tobacco3482 benchmark set, which is publicly available and contains 6,964 document images

belonging to various document types. All images were resized to 512 × 512 pixels, and treated under the same experimental conditions. The proposed framework has been realized in Python 3 using open-source libraries to enable reproducibility. Experimental results were achieved by running the authentication framework on the whole data set, involving normal verification, replay attack detection, ciphertext tampering analysis, watermark robustness evaluation, scalability analysis, statistical analysis, and computational complexity evaluation.

The cryptographic components have been programmed with the use of the Python cryptography library. It uses signature generation and verification via Ed25519, key exchange via X25519, session key derivation via HKDF and authenticated encryption via AES-GCM. Discrete Wavelet Transform (DWT)-based blind watermark embedding and extraction were done by using PyWavelets, and image preprocessing and attack simulation were performed by OpenCV. The following Python modules were used for numerical computation (NumPy), statistical analysis (Pandas & SciPy), and secure random number generation (secrets). The same parameter settings were used throughout all of the Tobacco3482 set for the JPEG compression, Gaussian noise, cropping, scaling and rotation attack experiments for the evaluation of watermark robustness, as well as for experiments on 10 to 2000 signatories for the evaluation of multi-signature scalability. The proposed framework can easily be reproduced by other researchers with a minimum amount of configuration, since all software libraries are open-source, publicly documented and broadly used by the research community.

Key Design Features of the Proposed Framework:

A number of cryptography-and-watermark based hybrid document authentication methods have been proposed, but many of the current methods pursue only a single security goal or a few. The authors argue in contrast, that multiple complementary security practices be embedded in a single authentication stream. The key characteristics of the proposed framework are summarized below.

XOR-based Multi-signature Aggregation: In the proposed framework, XOR-based key aggregation mechanism enables multi-signatory authentication to be scalable without incurring any significant communication overhead as compared to the independent verification of multiple signature from each participant.

Authenticated AES-GCM encryption: The framework uses AES-GCM authenticated encryption to give both confidentiality and authenticated transmission (and also ciphertext integrity).

Blind DWT watermarking: An extra forensic verification layer is used by embedding a blind discrete wavelet transform (DWT) watermark into each authenticated document. The watermark can be identified without the original document, allowing the system to still be useful in real-world applications for document verification.

Forward/reverse replay protection: Timestamp information is securely included in the encrypted authentication package. In a verification phase, timestamps are compared with a range of acceptable synchronization and replay attacks are therefore prevented, but a minor difference in the clocks of communicating entities can be tolerated.

Achieved unified verification architecture: Conventional schemes process cryptographic verifications and watermark authentication independently, while the proposed scheme handles all these processes in a unified verification workflow, which enhances the security and service quality.

Proposed Integrated Authentication Framework:

The cryptographic and watermarking framework designed in the proposed methodology in Figure 3 is an integrated model of authenticating signed documents by multiple signatories in a secure system. It is written in Python with publicly available, audited and popular scientific and cryptographic libraries to make it reliable and reproducible. It is implemented with the help of the cryptography library of elliptic curve cryptography, the

PyWavelets library of discrete wavelet transform, NumPy library of numerical operations and the hashlib library of secure hashing. The secrets module is used to create randomness needed in cryptographic operations, which consists of a cryptographically secure pseudo-random number generator, meeting modern security standards. They are open source and publicly available tools, i.e., other researchers can reproduce and extend the methodology without much configuration work. The proposed Scheme architecture is comprised of a number of interrelated stages such as key generation, watermark embedding, signature creation, authenticated encryption, verification, and attack evaluation. The properties of security each stage adds to the system are unique and together form a powerful mechanism of providing authenticity, confidentiality and tamper protection of documents.

Cryptographic key generation is the first step of the methodology as shown in Figure 3. During this step, public and private key pairs will be generated for each participant using Cryptographically Secure Pseudorandom Number Generator (CSPRNG) key generator. The private key will be randomly hashed using a 32 bytes value and public key will be obtained from the private key through elliptic curve multiplication. This process will ensure the one-way hash function property.

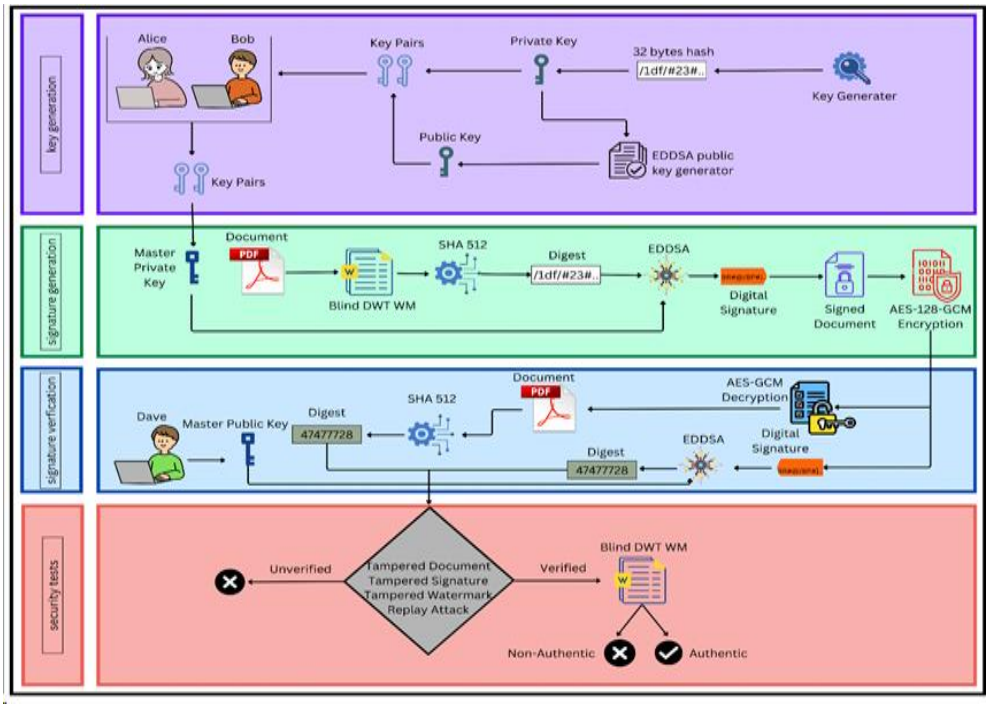

Figure 3. Research Architecture

Cryptographic Justification of XOR Based Key Aggregation:

The proposed framework is to combine the private key material of multiple signatories using bitwise XOR operation. This method allows for a lightweight aggregation mechanism that is linear in time complexity, O(n), which is efficient in terms of computation. Each participant generates a private key that is then added to the aggregated key; the aggregated key is dependent on all participants' private keys, which makes it hard to reconstruct without knowing all the private keys. The proposed aggregation is not for the purpose of replacing the standard threshold signature schemes, but it is for efficient document authentication and scalability.

The system is then used to combine all the individual private keys and public keys together to create one master private key and one master public key by a bitwise exclusive or (XOR) operation in order to have collaborative authorization between the various signatories. Mathematically, if there are n signatories with private keys k1, k2, kn and corresponding public

keys P1, P2, Pn, the aggregated master private key Kagg and the master public key Pagg are computed as:

$$K_{agg} = k_1 \oplus k_2 \oplus \cdots \oplus k_n \quad (1)$$

$$P_{agg} = P_1 \oplus P_2 \oplus \cdots \oplus P_n \quad (2)$$

This aggregation method will make sure every signatory makes a contribution to the overall signing authority and no one participant will be able to reconstruct the master key individually. They are the master private key and the master public key which are the result of the addition of the respective individual private and public keys respectively and the master public key is used later during verification. Besides keys Ed25519, an X25519 key pair is created to facilitate secure key exchange in the encrypted transmission. The Key Generation process is introduced in this Algorithm 1.

Algorithm 1 Key Generation
1: Input: Number of signatories n
2: Initialize: Aggregation buffer A with all zero bytes
3: Key Generation and Aggregation:
4: (SKi, PKi) ← Ed25519.KeyGen(), i = 1 . . . n
5: bi ← ExtractPrivateKeyBytes(SKi)
6: A ← A ⊕ b1 ⊕ b2 ⊕ · · · ⊕ bn
7: Derivation:
8: SKm ← DeriveMasterKey(A)
9: PKm ← ComputePublicKey(SKm)
10: Return: Master key pair (SKm, PKm)

Unlike conventional multisignature protocols that require iterative signature aggregation and multiple communication rounds, the proposed framework aggregates participant key material through a lightweight XOR operation, thereby reducing computational overhead while maintaining a compact authentication structure. The system goes into signature generation after key generation as shown in Figure 3. A forensic watermark is applied to the document before signing via a transform domain algorithm that is based on the Discrete Wavelet Transform (DWT). The document is first transformed into a matrix representation, and is broken down into four frequency sub bands LL, LH, HL, and HH in this process. The pattern of the watermark is created based on a pseudo-random sequence that is created using watermark metadata, like the identity of the signer and the time. This pattern is integrated into the LL sub-band with a controlled embedding factor, so that the watermark is not visible, and is resistant to compression and editing. Watermarking is embedded using the wavelet transform which is then inverted to recover the watermarked document.

A fixed length message digest is then generated by using the watermarked document with the SHA-512 cryptographic hash function.

$$H\text{=SHA-512(D)} \quad (3)$$

where D in equation (2) represents the document with a watermark. The digest (H) is the signature over the aggregated master private key by the EdDSA algorithm.

$$\sigma = \text{EDDSA Sign } (K_{agg}, H) \quad (4)$$

The deterministic nature of EdDSA guarantees that the same inputs will always generate the same signature, which erases vulnerabilities of nonce related schemes of randomized signatures. The produced signature, master public key, watermark metadata and timestamp are then encrypted into a structured data format. Algorithms 2 shows Blind DWT Based Watermark Embedding process.

Algorithm 2 Blind DWT-Based Watermark Embedding
1: Input: Document bytes D, watermark data W

2: Preprocessing:
3: I ← ConvertToGrayscale(D)
4: Wavelet Decomposition:
5: (LL, LH, HL, HH) ← DWT(I)
6: Watermark Embedding:
7: P ← GeneratePseudoRandomPattern(Hash(W))
8: LL' ← EmbedWatermark(LL, P, α)
9: Reconstruction:
10: Iw ← IDWT(LL', LH, HL, HH)
11: Dw ← ConvertToBytes(Iw)
12: Return: Watermarked document Dw

The system uses authenticated encryption with Advanced Encryption Standard (AES-128) in GCM to ensure the confidentiality of the documents transmitted. AES GCM also offers encryption and message authentication functions, making sure that any alterations to the ciphertext are detected during decryption. The signed document and metadata are encrypted using a 128-bit AES session key which has been randomly generated and the result is a ciphertext and authentication tag.

$$C = E_K\text{AES}(P) \quad (5)$$

where P in equation 3 represents the plaintext package.

To transmit the AES session key in a secure manner, the system wraps keys with X25519 elliptic curve Diffie Hellman key exchange and HKDF based key derivation. The sender generates a temporary key pair, calculates a shared secret with the public verifier key. The shared secret is processed using HKDF to derive a wrapping key that encrypts the AES session key. The eventual message sent is thus the ciphertext, authentication tag, the wrapped AES key, and the temporary public key of the sender illustrated in Figure 3. Such a stratum of encryption will provide a secure distribution of keys and safeguard the data and communication channel against interception. Signature Generation process is presented in Algorithm 3.

Algorithm 3 Signature Generation

1: Input: Watermarked document Dw, master private key SKm
2: Hashing:
3: h ← SHA-512(Dw)
4: Signature Generation:
5: σ ← EdDSA.Sign(h, SKm)
6: Encryption Preparation:
7: KAES ← RandomKey(128)
8: M ← CreateMetadata(timestamp, W)
9: Payload Encryption:
10: C ← AES-GCM.Enc(KAES, Dw M)
11: Key Wrapping:
12: (ske, pke) ← X25519.KeyGen()
13: S ← X25519.Derive(ske, PKr)
14: Kw ← HKDF-AES-GCM(S, KAES)
15: Packaging:
16: P ← (C, Kw, pke, σ)
17: Return: Encrypted signature package P

The verification step does the reverse to verify the authenticity of the documents. The verifier retrieves the AES session key based on X25519 (key exchange) and then decrypts the payload with AES-GCM shown in Figure 3. Any change to the encrypted data causes

authentication to fail during decryption. Upon successful decryption of the embedded timestamp, it is checked against a predefined replay window.

$$|t_{\text{current}} - t_{\text{embedded}}| \leq \Delta t_{\max} \quad (6)$$

In case the timestamp is out of range, then the document is rejected as a replayed message.

The hash is then calculated again (SHA-512) on the recovered document and compared to the stored digest to check integrity. The signature of the EdDSA is then verified with the aggregated master public key shown in Figure 3 research architecture diagram. After the signature authenticity has been verified the embedded watermark is removed and matched with the reproduced pseudo-random watermark pattern based on the seed stored. When the correlation coefficient is more than a predefined threshold, it is said to be valid and the document content has not been modified. The document is accepted upon complete verification steps decryption, replay validation, signature verification and watermark correlation being successfully met. Algorithm 4 presents Verification process.

Algorithm 4 Verification

1: Input: Encrypted package P
2: Sniffing Attack Resistance:
3: Assume adversary may capture P over an insecure channel
4: Verify payload confidentiality through authenticated encryption
5: Decryption without valid key fails
6: Key Recovery:
7: KAES ← RecoverKeyUsingX25519(P)
8: Decryption:
9: (Dw, h, t, W) ← AES-GCM.Dec(KAES, P)
10: Replay Protection:
11: Verify t is within the allowed time window
12: Integrity Verification:
13: h' ← SHA-512(Dw)
14: Check h’ = h
15: Watermark Verification:
16: ϱ ← ComputeCorrelation(ExtractWatermark(Dw), W)
17: Decision:
18: Accept document if all verifications succeed
19: Reject document otherwise

To determine the strength and performance of the proposed scheme, simulations of various tampering conditions, such as modifications to the content of the document, to the digital signature, to the watermark data and to the timestamps, are performed in order to evaluate the effectiveness of tampering detection and to prevent replay attacks. All modified packages are run through the verification algorithm to make sure that tampering can be reliably detected can be seen in Figure 3. In parallel, the performance in terms of computation is recorded, and the time taken to generate key, create signature, and verify is noted. This benchmarking offers quantitative data about the effectiveness, scaling capacity, and safety of the suggested scheme. Key generation, signature creation, encryption and verification execution times are measured in several iterations. The findings revealed that the system sustains sub-second processing time of most text and PDF documents in multi-signatory systems. Key aggregation via XOR is known to heavily minimize the amount of computational overhead in comparison with the more traditional multi-party signature schemes, and deterministic EdDSA signing provides efficient verification. Algorithm 5 shows the Tampering Detection and Performance Benchmarking.

Algorithm 5 Tampering Detection and Performance Benchmarking

```
Input: Encrypted package P
Tampering Evaluation:
Modify document content and verify tampering detection
Modify digital signature and verify tampering detection
Modify watermark data and verify tampering detection
Modify timestamp to simulate replay attack
Performance Measurement:
tkey ← MeasureKeyGenerationTime()
tsig ← MeasureSignatureGenerationTime()
tver ← MeasureVerificationTime()
Output:
Record tampering detection results and average execution times
```

Result and discussion:

Performance Evaluation of the Proposed Authentication Framework:

This part is an analysis of the computational efficiency of the suggested digital document authentication scheme. In the performance analysis, the emphasis is made on the key cryptographic operations of the authentication process, such as the key generation, signature generation, signature verification, and the overall execution time. These indicators will be vital in establishing the scalability and the applicability of the system in the real-world digital document authentication context.

The experiment was done by changing the number of signatories between 10 and 500 to simulate more workloads in collaborative document authentication systems. Each cryptographic operation was timed in seconds. To illustrate the efficiency of the given approach, the outcomes were contrasted with various existing methods that are reported in the literature, such as the schemes suggested by [16], [17], [18], [15], and [19]. Table 1 provides the detailed numerical results, whereas the graphical comparisons are shown in Figure. 4(a)-(d).

Key Generation Time Analysis:

The first step of the digital signature process is key generation. The proposed scheme proves to be much faster in key generation time than the current methods as can be seen in Table 1 and Figure. 4(a). To demonstrate this, the execution time of the proposed scheme is about 0.07595 seconds when we have 500 signatories, compared with much longer execution times in the methods of [16][17] and [18]. This improvement is mainly due to the efficient key aggregation mechanism and the use of lightweight CSPRNG key generator used in the proposed scheme.

Signature Generation Time Analysis:

The main process of the authentication system is signature generation. As demonstrated by the values in Table 1 and Figure. 4(b), it is clear that the proposed scheme has the lowest signature generation time at all the workloads tested. The proposed method takes about 0.05283 seconds at 500 signatories, whereas the compared methods have higher computational overhead. The efficiency is mainly attributed to the deterministic signing process and optimized cryptographic operations and the use of multi-signature technique.

Table 1. Performance Comparison of Digital Signature and Authentication Schemes

Method	Parameters	10	50	100	200	500
[16]	Key Generation	0.01812	0.07214	0.13925	0.28351	0.72136
	Signature Generation	0.00987	0.04163	0.08312	0.16042	0.40219
	Signature Verification	0.01345	0.05671	0.11594	0.23065	0.59032
	Overall	0.04144	0.17048	0.33831	0.67458	1.71387
[17]	Key Generation	0.01594	0.06855	0.14582	0.30112	0.78214
	Signature Generation	0.00812	0.03379	0.07132	0.14317	0.36128
	Signature Verification	0.01021	0.04013	0.08291	0.16354	0.41196
	Overall	0.03427	0.14247	0.30005	0.60783	1.55538
[18]	Key Generation	0.02245	0.07918	0.16033	0.32516	0.89471
	Signature Generation	0.01194	0.04320	0.09056	0.18714	0.46337
	Signature Verification	0.01483	0.05891	0.12015	0.24381	0.58264
	Overall	0.04922	0.18129	0.37104	0.75611	1.94072
[15]	Key Generation	0.01231	0.05418	0.11826	0.24119	0.65523
	Signature Generation	0.00692	0.02744	0.05732	0.11973	0.31248
	Signature Verification	0.00911	0.03683	0.07951	0.16124	0.39897
	Overall	0.02834	0.11845	0.25509	0.52216	1.36668
[19]	Key Generation	0.02851	0.10165	0.19642	0.38329	1.01255
	Signature Generation	0.01582	0.06171	0.12814	0.25194	0.67919
	Signature Verification	0.02213	0.09103	0.19284	0.37941	0.99811
	Overall	0.06646	0.25439	0.51740	1.01464	2.68985
Proposed Scheme	**Key Generation**	**0.00192**	**0.00759**	**0.01517**	**0.03043**	**0.07595**
	Signature Generation	**0.00112**	**0.00535**	**0.01056**	**0.02117**	**0.05283**
	Signature Verification	**0.00123**	**0.00609**	**0.01213**	**0.02455**	**0.06124**
	Overall	**0.00428**	**0.01901**	**0.03787**	**0.07615**	**0.19002**

Signature Verification Time Analysis:

Each time a document is authenticated signature verification is carried out. As Table 1 and Figure. 4(c) demonstrate, in the proposed scheme, the verification time is much faster than that of the current techniques. The proposed system takes about 0.06124 seconds to verify 500 signatories, whereas the rival algorithms take significantly longer. Such enhancement is achieved due to an aggregated verification mechanism as the verification is done in a single step even if there are multiple signatories unlike other techniques which follow serial flow of verification, and the proposed scheme also lessens the number of cryptographic calculations.

Overall Execution Time Analysis:

The total execution time is the total cost of key generation, signature generation and verification. As indicated in Table 1 and Figure 4 (d), the proposed scheme has the lowest overall execution duration as compared to all other methods. The time taken by the proposed method is about 0.19002 seconds when the number of signatories is 500 and the schemes currently in use take much more time to execute.

This efficiency is achieved mainly due to the application of the EdDSA digital signature scheme, which by default demands much smaller keys and signature sizes than traditional algorithms, like RSA and DSA, hence lowering the computational and communication overhead. Also, a multi-signature technique is very important to be integrated, rather than producing and checking individual signatures of each party, the proposed scheme combines all the signatures into a single compact signature. This aggregation eliminates the cryptographic operations by a large factor and the overall amount of data sent to the destination hence a near constant verification time independent of the number of signatories which is also discussed in the subsequent section.

The experimental findings show that the suggested scheme is more computationally efficient and scalable than the currently existing digital document authentication methods and therefore it can support real time secure document verification systems.

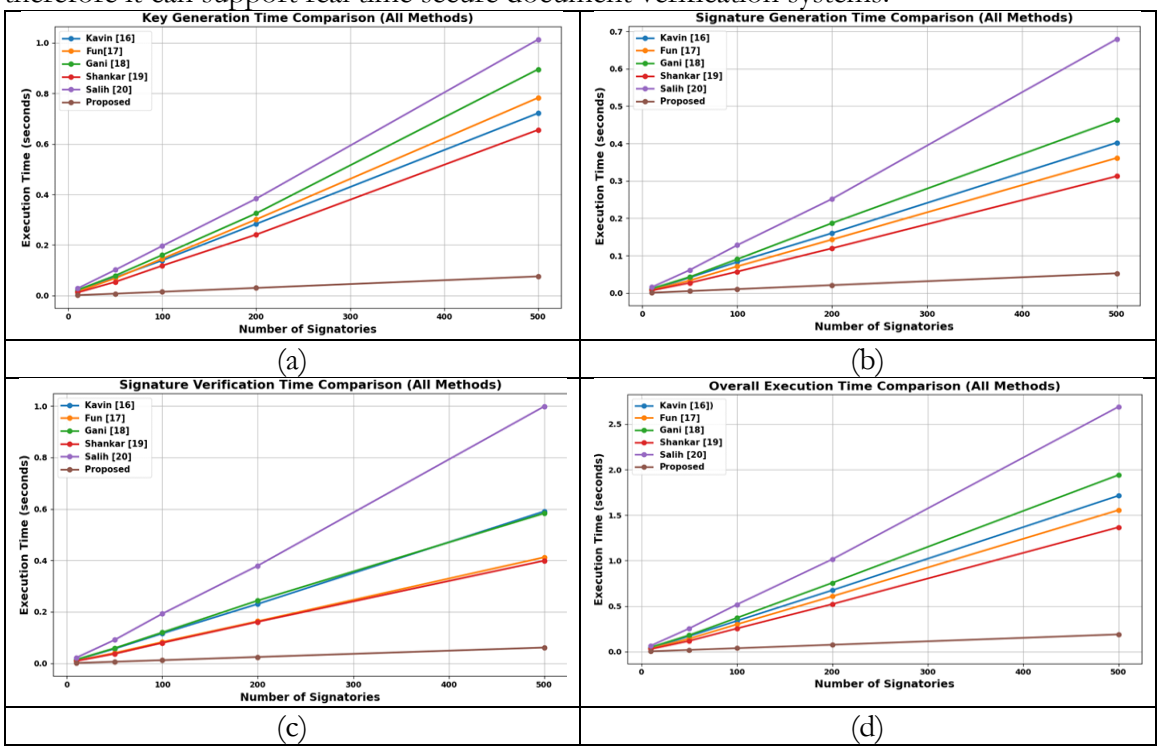

Figure 4. Comparative performance evaluation of (a) key generation time, (b) signature generation time, (c) signature verification time, and (d) total execution time across varying numbers of signatories.

Communication and Storage Cost Analysis:

The suggested multi-signature scheme has the same communication and storage cost of 64 bytes, irrespective of the number of signatories involved (see Table 2). This is the non-variable cost that is linked with a single signature and single public key generated by the multi-signature. In comparison, traditional non-multi-signature has a linear increment of both the cost of communication and storage, as each party adds an independent signature and public key.

To illustrate, the number of signatories is 10: in the non-multi-signature scheme, 640 bytes are needed, in the multi-signature scheme, just 64 bytes are needed, and this translates to a 90 percent reduction in both communication and storage overheads. The decrease is much more pronounced with more signatories. The reduction in communication and storage is 99% with 100 signatories and 99.8 with 500 signatories.

This trend is a clear indication of scalability advantage of the proposed multi-signature scheme. The proposed scheme is an effective way to remove bottlenecks in communication and storage overhead in large collaborative systems, e.g. organizational approval systems, distributed authorization workflows, and multi-party contract validation systems, which can often occur in traditional digital signature architectures.

Verification Cost and Speedup Analysis:

Verification efficiency is a very important performance measure of digital authentication systems, especially when documents have to be verified by more than one participant. As shown in Table 2, multi-signature scheme is always much faster than non-multi-signature scheme in terms of verification time.

As an example, the multi-signature scheme verification time of 10 signatories is 0.000545 seconds, compared to 0.003256 seconds by the non-multi-signature scheme, and a 5.97x speedup is achieved. The larger the number of signatories, the larger is the performance gap. The verification speedup is over 60 times faster at 50 signatories, and is more than 250 times faster at 200 signatories. Of particular importance, at a signatory number of 500, the suggested multi-signature scheme provides a speedup of over 1160 times in the verification.

The reason the multi-signature technique gets speedup performance is mainly due to the aggregation of multiple signatories into a single compact signature and merged verification process. In non-multi-signature techniques, each signatory generates an independent signature that needed to be verified separately which causes verification cost to grow linearly with the number of signatories. For example, 100 signatories require 100 separate verification operations. On the other hand, the proposed multi-signature technique combines all signatories into a single final signature associated with aggregated public key. As a result, only one verification process is required regardless of the number of signatories which leads to major reductions in computational overhead and verification time. Therefore, as the number of signatories increases, the performance advantage of the multi-signature scheme becomes even more significant.

Figure 5 shows a logarithmic scale comparison of the cost of verification of both schemes. It is clearly illustrated in the graph that the time taken to verify under the non-multi-signature scheme is growing at an objective description because of the necessity to verify each signature separately. Conversely, the multi-signature scheme has a nearly fixed cost of verification, as only one aggregate signature has to be verified.

This finding validates the hypothesis that suggested multi-signature method makes a significant improvement in terms of efficiency in verification and the complexity of verification is almost a constant with regards to the number of signatory parties. The suggested scheme is therefore very appropriate in large scale collaborative authentication systems where speed of verification and minimization of resources is of great importance.

Table 2. Enhanced Quantitative Comparison of Multi-signature vs Non-Multi-signature Schemes Across Signatories

Signatories	Comm. Multi (B)	Comm. Non-Multi (B)	Reduction (%)	Storage Multi (B)	Storage Non-Multi (B)	Reduction (%)	Verify Multi (s)	Verify Non (s)	Speedup (×)
10	64	640	90.00	64	640	90.00	0.000545	0.003256	5.97
30	64	1920	96.67	64	1920	96.67	0.001040	0.012029	11.56
50	64	3200	98.00	64	3200	98.00	0.000235	0.014111	60.04
100	64	6400	99.00	64	6400	99.00	0.002247	0.031012	13.80
200	64	12800	99.50	64	12800	99.50	0.000231	0.058472	253.11
500	64	32000	99.80	64	32000	99.80	0.000252	0.293109	1163.92

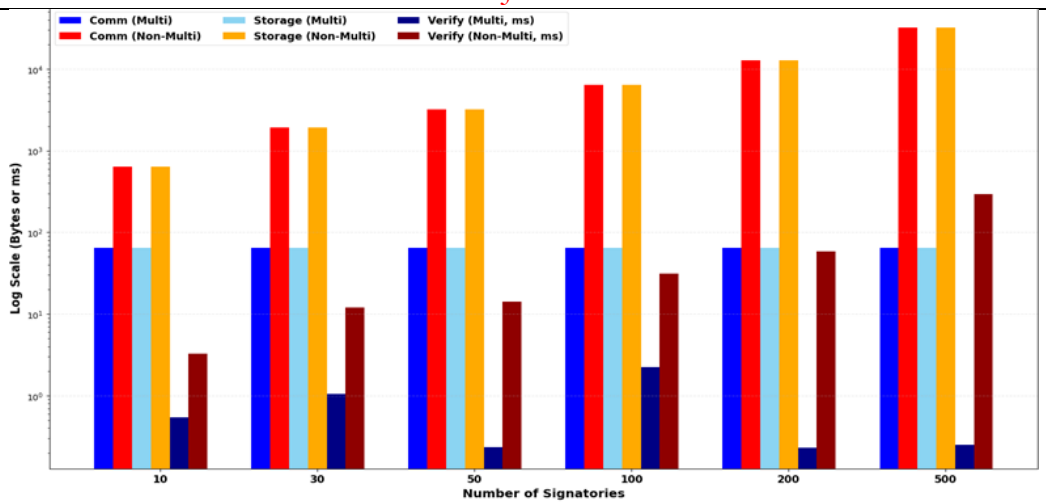

Figure 5. Log-Scale Cost Comparison of Multi-signature and Non-Multi-signature Schemes

Execution Analysis of Watermarking Encryption:

Two configurations, watermarking encryption authentication and no watermarking encryption authentication were compared to measure the computational impact of watermark-based verification being implemented in the document authentication pipeline. The former setup incorporates DWT founded watermark embedding and correlation checking along with cryptographic signing and encryption, but the latter setup incorporates only cryptographic signature checking. The measured execution times are summarized in Table 3.

Table 3. Execution Time Comparison with and without Watermarking Encryption

Signatories	With Watermarking Encryption (s)	Without Watermarking Encryption (s)
1	0.0045009	0.0008686
2	0.0045738	0.0009505
3	0.0048027	0.0010440
4	0.0048503	0.0011630
5	0.0049185	0.0012340
10	0.0075566	0.0017406
20	0.0065450	0.0033295

Table 3 results indicate that the moderate computational overhead of watermarking addition is added to the non-watermarked authentication procedure. To illustrate, with a single signatory configuration the watermarking encryption process takes about 0.0045 seconds to complete the authentication process, compared to 0.0008686 seconds to complete the same process without watermarking encryption. The extra processing time is mainly due to the implementation of DWT decomposition, embedding watermarks coefficients, reconstructing the document and verifying the correlation in the authentication process.

The total execution time of both layouts grows slightly as the number of signatories grows, but the expansion is not chaotic and is predictable. The watermarked authentication process takes about 0.0049 seconds at the five signatories as opposed to 0.001234 seconds in the non-watermarked setup. Although the watermarking method always takes a longer time to process, this difference is always in a stable range. This shows that the additional overhead caused by watermark insertion and removal is not highly dependent on the number of signatories, and is largely determined by the cost of processing in the transform-domain. This comparison is plotted in a graphical form as illustrated in Figure 6.

Figure 6 shows that the execution time when the watermarking is encrypted is always higher than the non-watermarked one at all levels of the signatures. The gap between the two curves, however, does not soar as the number of signatories increases. Rather, the

performance does not decrease as the workload of authentication increases, but instead the performance is still computationally stable. This result shows that the watermarking system does not add an exponentially growing processing load but a fixed processing load.

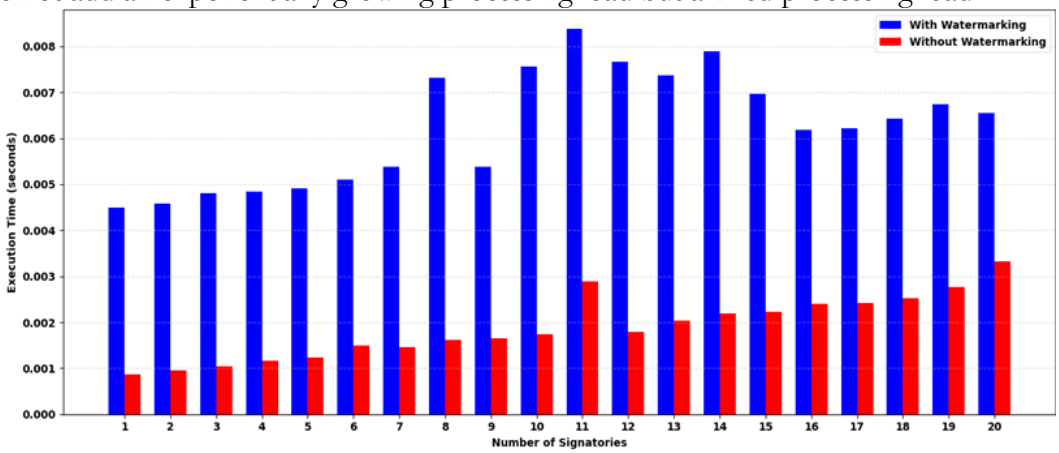

Figure 6. Execution Time Comparison with and Without Watermarking Encryption

Although the watermark technique introduces some computational overhead, it greatly enhances the authentication of documents by adding another verification layer to the cryptographic signature verification. The experimental robustness verification shows that the embedded watermark can survive the compression, noise, cropping, scaling attacks, and enhances the robustness of the document to manipulation and unauthorized modification. The current implementation using DWT is sensitive to the geometric rotation in the document, but it can be used for reliable authentication for some of the most commonly used document processing operations during storage and transmission.

Watermark Robustness Evaluation:

The proposed blind DWT watermarking scheme was tested using various image processing attacks such as JPEG compression, Gaussian noise addition, cropping, scaling, and rotation in order to assess the robustness of the proposed scheme. The attacks were performed on the authenticated document and the embedded watermark was extracted and checked using the correlation-based detection algorithm. The performance was evaluated by Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM), watermark correlation coefficient, and watermark detection success.

The experimental outcomes show that the proposed watermarking technique is able to achieve good detection performance not only when the images are JPEG compressed, but also when images are corrupted by additive Gaussian noise, cropped and scaled. The watermark detection process was successful for all the compression qualities (30–100), Gaussian noise (σ = 5–20), cropping percentage (5–30%) and scaling factor (0.5×–2×). As the attack severity increased, the watermark correlation also began to drop, but it is still above the predefined detection threshold, giving a 100% detection rate for these attack categories.

Geometric rotation, on the other hand, was found to have a more significant effect on watermark synchronization. With the rotation angles of 5°, 10°, 20°, and 30°, the correlation coefficient was below the detectable level, and the watermark could not be detected. This behavior was anticipated as it is a natural property of the conventional DWT-based blind watermarking technique, which is sensitive to spatially changing relative wavelet coefficient positions.

Table 4. Watermark Robustness Evaluation

Attack	Parameters	Detection Rate	Observation
JPEG Compression	Quality 30–100	100%	Robust
Gaussian Noise	σ = 5–20	100%	Robust

Cropping	5–30%	100%	Robust
Scaling	0.5×–2×	100%	Robust
Rotation	5°–30°	0%	Sensitive
Overall	24 attack cases	**83.33%**	Average detection rate

Cost Categorization of the Proposed Scheme:

In order to assess system performance further, a classification model was constructed to classify the computational cost, communication overhead, and signature size into three categories, namely, high, medium, and low. These thresholds present a unified system of understanding the performance of the proposed scheme when compared to known systems of digital signature. Table 4 shows the classification criteria.

Table 4. Criteria for Computational and Communication Cost Categorization

Parameter	High	Medium	Low
Computational Cost (ms)	> 6 ms	3–6 ms	≤ 3 ms
Communication Cost (bytes)	> 120 bytes	81–120 bytes	≤ 80 bytes
Signature Size (bytes)	> 64 bytes	49–64 bytes	≤ 48 bytes

The computational cost classification is an indication of the overall processing time needed to generate keys, embed watermark, generate digital signature, encrypt, and verify. Execution times less than three milliseconds are therefore often ignored and labeled as low computational cost in modern cryptographic systems in the real-time applications. Execution times of three to six milliseconds are classified as medium cost, which is moderate computational cost of layered security architectures. A processing time longer than six milliseconds is high cost.

Communication cost is the data sent over in document authentication which includes digital signatures, encrypted payloads, public keys and watermark metadata. Any communication overhead less than eighty bytes is classified as low cost which implies that the bandwidth usage is low enough to be used in a distributed or bandwidth-limited environment. Transmission sizes between eighty-one and one hundred and twenty bytes reflect medium overheads with higher values indicating high communication cost normally experienced with traditional multi-signature concatenation strategies.

The size of signature is also a factor in the efficiency of the system since a larger signature increases storage and communication needs. The size of a typical standard EDDSA signature is sixty-four bytes, but optimized aggregate signatures can be much shorter. The architecture proposed makes use of signature aggregation methods to reduce signature growth, thus ensuring a small signature footprint that can be used in scalable multi-signatory systems.

Computational Complexity Analysis:

Table 5. Computational Complexity of the Proposed Framework

Operation	**Complexity**	**Remarks**
SHA-512 Hashing	O(n)	n is document size
AES-GCM Encryption/Decryption	O(n)	Linear with document size
X25519 Key Exchange	O(1)	Constant-time elliptic-curve operation
Ed25519 Signature Generation	O(1)	Constant-time signing
Ed25519 Signature Verification	O(1)	Constant-time verification
XOR Key Aggregation	O(k)	k is the number of signatories
DWT Watermark Embedding	O(n)	Linear image transform
DWT Watermark Detection	O(n)	Linear correlation-based detection

Computational complexity of the proposed authentication framework is summarized in Table 5, where the hash computation, authenticated encryption and embedding of watermark in the DWT has linear time complexity with respect to document size (n). The multi-signature key aggregation based on XOR requires O(32k) = linear complexity with

respect to the number of signatories (k), with one XOR operation per signatory's private key. Ed25519 signature generation and verification, and the X25519 key exchange are elliptic-curve operations that are constant time. As such, the overall complexity of the framework is O(n + k), where n is the size of the document and k is the number of signatories. This complexity shows that the proposed framework is efficient in terms of the number of documents and that it has a linear overhead on the number of signatories.

Formal Security Analysis:

The security of the proposed authentication framework is based on the following security properties of the individual components: SHA-512 hash, Ed25519 digital signatures, AES-GCM authenticated encryption, blind DWT watermarking, and replay protection using timestamps. The document integrity is assured since any alteration of the document will alter the document's SHA-512 hash, and thus prevent the signature to be verified. Moreover, the embedded blind DWT watermark is disturbed by any changes to the document content, which enables content-level tampering detection without cryptographic verification. To successfully authenticate, both the digital signature and watermark must be authenticated successfully, thus two complementary layers of authentication are provided.

Signer authenticity is provided by the Ed25519 digital signature scheme, which only people with valid private keys can create acceptable signatures for. This proposed XOR based key aggregation is done locally in key generation phase and the aggregated key is never sent through the communication channel. So, recovering the aggregated signing key will need access to all the private key components and the aggregation process will add only linear computational overhead with the number of signatories. AES-GCM authenticated encryption ensures confidentiality and message integrity by preventing the authentication metadata from being stolen or altered during transmission. Furthermore, replay attacks are prevented by including a timestamp into the encrypted authentication package and checking the timestamp at the time of verification. The verifier then uses a programmable amount of time tolerance window with respect to the other time tolerance window to cover practical deployment scenarios where the other time tolerance window may differ slightly from the one from the communicating devices in regard to the clock synchronization or the network delay. Documents/packages with timestamp within the acceptable limit are allowed to proceed while the ones with the expired or the delayed timestamp are rejected as replay attempts. For systems with more demanding timing needs, conventional clock synchronization methods like the Network Time Protocol (NTP) can be used to keep systems "in sync" among the participating devices. The security features work together to prevent that a document is tampered with, a signature is forged, a replay attack, a man-in-the-middle attack, or unauthorized changes are made, while simultaneously offering efficient multi-signature authentication for large-scale document verification.

Statistical Analysis:

To assess the reliability and consistency of the proposed authentication framework, a statistical analysis was carried out with the full test set of document images containing 6,964 images. The main performance metrics were calculated descriptively using the mean, standard deviation and 95% confidence interval. These numbers offer a measure of the robustness and reliability of the proposed scheme in large-scale experiments.

Table 6. Statistical Analysis of Authentication Performance

Metric	Mean	Std. Dev.	95% Confidence Interval
Signing Time (s)	0.003583	0.000511	0.003570–0.003595
Verification Time (s)	0.002462	0.000468	0.002451–0.002473
Communication Cost (Bytes)	232	0	232–232
Storage Cost (Bytes)	232	0	232–232

Table 6 shows the statistical properties of the proposed authentication mechanism when applied to the entire benchmark set. The average signing and verification times were 0.003583 s and 0.002462 s, respectively. Small standard deviations (0.000511 s for signing and 0.000468 s for verification) show that the execution time was very stable throughout all 6,964 document images. Also, the narrow 95% confidence intervals indicate that there is low variability and high repeatability of the performance measured. The communication and storage costs were fixed as 232 bytes, and had no variation in any of the experiments. The results show that the proposed framework could well be considered in terms of delivering stable, predictable and reliable performance for the large-scale document authentication task.

Security Comparison with Existing Authentication Schemes:

In an attempt to test the effectiveness of the proposed framework and the reference methods were compared on the same basis of performance that has been reported in the literature, such as: communication overhead, signing time, verification time, replay protection and security functionality. The proposed implementation was performed on the same experimental conditions for all the benchmark images and the performance of the baseline methods was compared based on their published characteristics. The comparison focuses on algorithmic efficiency and security features and not on the execution time which depends on the hardware. Additionally, all the cryptographic primitives used in the proposed framework are standardized algorithms (Ed25519, X25519, AES-GCM, and SHA-512), which means that the comparison is between how these algorithms are implemented versus optimizations specific to the architecture.

Table 7. Comparison of Existing Schemes vs Proposed Multi-signature Watermarking Encryption Model

Scheme	Multi-Signature	Blind DWT Watermark	Replay Protection	Unified Verification	Comm. Cost	Comp. Cost	Security
[16]	✗	✗	✗	✗	Low	High	Medium
[17]	✗	✗	✗	✗	Medium	High	Medium
[18]	✗	✓	✗	✗	Medium	High	High
[15]	✓	✗	✗	✗	Medium	Medium	High
[19]	✗	✓	✗	Partial	High	High	High
Proposed Framework	✓	✓	✓	✓	Low	Low	**Very High**

The proposed framework demonstrates a good balance between security and computational efficiency, as seen in Table 7. Its communication overhead is still 232 bytes, more than a signature generated with the standalone Ed25519, but contains additional security features, such as authenticated encryption, blind watermark checking, replay protection, and unified verification, which are not supported in a combined way by the compared schemes. As such, it is a fair comparison, since it compares entire authentication systems with the security features they provide, instead of comparing individual cryptographic primitives.

The suggested multi-signature watermarking encryption model overcomes these limitations by a multilayered security infrastructure. The integration of EdDSA based digital signatures guarantees authenticity and integrity of the document content. AES GCM authenticated encryption guarantees the confidentiality of the transmitted data and a built-in integrity check using authentication tags. Moreover, the blind DWT watermarking can insert signer specific identifiers in the document text, which means that even with the loss of cryptographic credentials, forensic level detection of tampering is possible.

Figure 7 graphical analysis shows that the majority of existing schemes only deal with security threats one by one, but not a multi-faceted scheme of defense. Conversely, the

proposed scheme offers safeguards against MIM, replay, and sniffing at the same time. This combined defence mechanism greatly boosts the security of the multi-signatory digital document authentication.

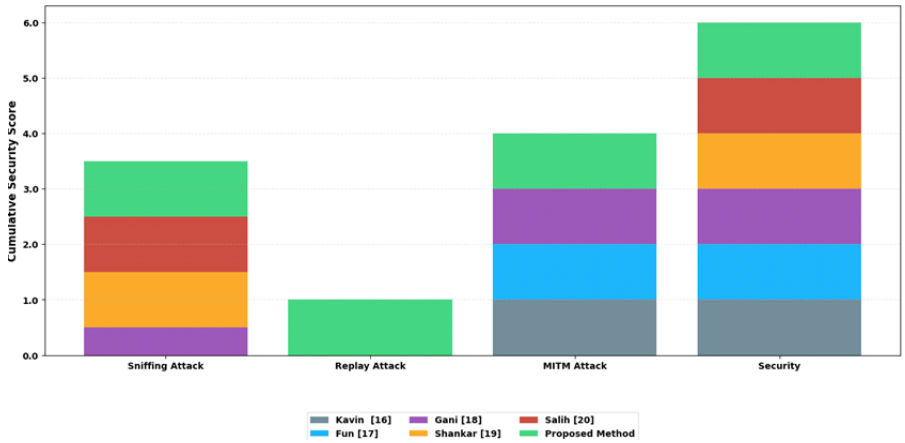

Figure 7. Provides a graphical comparison of the security coverage offered by existing authentication schemes and the proposed model.

Overall Discussion:

The overall outcomes of the experiment prove that the proposed scheme can provide a good balance between the security robustness and the computational efficiency. Although the overhead of the watermarking integration is higher than that of simply signature based authentication schemes, the absolute execution time is incredibly low and would lie far below the acceptable threshold of a real-time digital document processing system.

Among the main benefits of the suggested scheme is that it can be scaled in many signatory scenarios. This scalability is achieved because XOR-based key aggregation introduces only linear computational overhead with respect to the number of signatories while preserving a fixed-size aggregated signing key. Signature aggregation is much more economical in communication overhead and complexity in verification than the traditional methods in which each signatory supplies a distinct signature. Also, the watermarking mechanism adds a separate verification channel to improve the verification of document authenticity on the content level.

Nevertheless, there are some limitations to the scheme. The added watermark embedding and extraction functions add to the computational complexity in comparison to a minimal signature-only system. This extra overhead can be optimized by hardware acceleration or other optimized wavelet methods in very resource-constrained environments. Although this is limiting, the execution time that is observed is low enough to be practically applicable in most real-world applications involving digital authentication. The findings verify that the suggested multi-signature watermarking encryption structure offers a higher level of security, scalable authentication speed, and efficient tamper detection functionalities. Combining cryptographic authentication, encrypted communications, and transform-domain watermarking creates a strong architecture that can be used to facilitate the current security needs of legal, financial, and forensic documents in legal, financial, and forensic settings.

Conclusion and Future Work:

This paper introduces a unified hybrid authentication framework for secure digital document verification by combining XOR-based multi-signature aggregation, Ed25519 digital signatures, AES-GCM authenticated encryption, blind DWT watermarking and replay-resistant verification within a single authentication architecture. The proposed framework differs from the existing hybrid authentication schemes in that it offers authentication, confidentiality, integrity, forensic traceability, replay protection and scalable multi-signatory verification in a single framework, while not requiring high levels of computation.

Experimental results on a test set comprising 6,964 document images show that the proposed framework can sign and verify a document at an average rate of 279.13 documents/s and 406.13 documents/s respectively. The stability of the proposed approach is also supported by a statistical analysis, which shows a very small standard deviation of 0.000511 s for signing and 0.000468 s for verification, respectively, as well as very narrow 95% confidence intervals. The scalability experiments also show that the framework works well for up to 2000 signatories with an almost constant verification time, and a fixed communication and storage overhead of 232 bytes.

The watermark robustness evaluation shows reliable authentication performance against JPEG compression, Gaussian noise, cropping and scaling attacks with an average watermark detection rate of 83.33%, with rotation attack being the most difficult one. By combining the use of authenticated encryption, a document timestamp verification, and a blind watermark verification, the security experiments further demonstrate the resistance of the proposed framework against document tampering, replay attacks, sniffing, and man-in-the-middle attacks. The findings of this study demonstrate that the proposed approach is able to achieve a good balance of security, computational efficiency, scalability and forensic traceability of digital documents for legal, financial, governmental and enterprise applications.

In future, more research will be devoted to increasing the robustness of the system against geometric attacks, studying post-quantum algorithms for digital signatures to ensure resilience to emerging quantum threats, ensuring an immutable audit trail with blockchain-based audit logging, and providing lightweight implementations for low-power IoT devices and edge computing applications.

Acknowledgement:

The author acknowledges the University of Peshawar who contributed to this research by giving technical resources, supervision and access to computational facilities.

Conflict of Interest:

The authors state that they have no conflict of interest when it comes to the publication of this research paper.

References:

[1] Cyprian Omukhwaya Sakwa, Fagen Li., "Survey on post-quantum cryptography implementations and deployment challenges," *Comput. Sci. Rev.*, vol. 61, 2026, doi: https://doi.org/10.1016/j.cosrev.2026.100975.

[2] G. Kumar, B. R. Killi, A. K. Das, and Y. Park, "Post-Quantum Secure Lattice-Based Lightweight Authentication and Key Agreement Scheme for Multi-Layer IoT-Enabled Smart Grid System," *IEEE Internet Things J.*, 2026, doi: 10.1109/JIOT.2026.3673083.

[3] "Digital government index: Government at a Glance 2025 | OECD." Accessed: Jul. 12, 2026. [Online]. Available: https://www.oecd.org/en/publications/2025/06/government-at-a-glance-2025_70e14c6c/full-report/digital-government-index_1edec44e.html

[4] Farkhund Iqbal, Aasia Jaffri, "Forensic investigation of small-scale digital devices: a futuristic view," *Front. Commun. Networks*, vol. 4, 2023, doi: https://doi.org/10.3389/frcmn.2023.1212743.

[5] Abdul Basit Shahid, Khwaja Mansoor, Yawar Abbas Bangash, "Post-quantum cryptographic authentication protocol for industrial IoT using lattice-based cryptography," *Sci. Rep.*, 2026, [Online]. Available: https://www.nature.com/articles/s41598-025-28413-8

[6] . Mariya Thomas, Mahima Mary Mathews, Panchami V., Asha Sebastian,Sivaraman Eswaran, "Securing the future: A comprehensive review of post-quantum digital signatures," *Comput. Sci. Rev.*, vol. 61, 2026, doi:

https://doi.org/10.1016/j.cosrev.2026.100935.

[7] Carsten Baum,Bernardo David Elena Pagnin, Akira Takahashi., "Hedged ECDSA and EdDSA Signatures," 2024, [Online]. Available: https://www.ietf.org/archive/id/draft-irtf-cfrg-det-sigs-with-noise-03.html#name-introduction

[8] Anmol Singh,Tanishque Sharma., "AES vs AES_GCM for Data Protection: A Comprehensive Security Comparison," *Reseachgate*, 2024, doi: 10.1109/OTCON60325.2024.10688085.

[9] "(PDF) Efficient Key-Based Encryption and Authentication for Advanced Digital Forensic Storage Security." Accessed: Jul. 12, 2026. [Online]. Available: https://www.researchgate.net/publication/392325162_Efficient_Key-Based_Encryption_and_Authentication_for_Advanced_Digital_Forensic_Storage_Security

[10] C. Baum, B. David, E. Pagnin, and A. Takahashi, "Universally Composable Interactive and Ordered Multi-Signatures," *Cryptol. ePrint Arch.*, vol. 15675 LNCS, pp. 3–31, 2025, doi: 10.1007/978-3-031-91823-0_1.

[11] S. M. H. Saif Aldeen S. Naem, "Digital watermarking techniques, challenges, and applications: A review," *Mesopotamian J. CyberSecurity*, vol. 5, no. 2, pp. 1–24, 2025, doi: 10.58496/MJCS/2025/028.

[12] Shahzad Ahmad, Stefan Rass, Zahra Seyedi, "Post-Quantum Sanitizable Signatures from McEliece-Based Chameleon Hashing," *arXiv:2602.20657*, 2026, [Online]. Available: https://arxiv.org/abs/2602.20657

[13] . Ferik B, Laimeche L, "An Efficient Semi-blind Watermarking Technique Based on ACM and DWT for Mitigating Integrity Attacks," *Arab. J. Sci. Eng.*, vol. 50, pp. 15885–15905, 2025, doi: https://doi.org/10.1007/s13369-025-09992-0.

[14] . Nova Rijati, Sudipta Kr. Ghosal, Aditya Kumar Sahu, "DWT- DCT Image Watermarking with Quantum-Inspired Optimization," *Int. J. Intell. Eng. Syst.*, vol. 18, no. 1, 2024, doi: 10.22266/ijies2025.0229.74.

[15] Gauri Shankar, Liwa H. Ai-Farhani, P. Anitha Christy Angelin, "Improved Multisignature Scheme for Authenticity of Digital Document in Digital Forensics Using Edward-Curve Digital Signature Algorithm," *Secur. Commun. Networks*, 2023, [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1155/2023/2093407

[16] . Balasubramanian Prabhu Kavin, Ganapathy Sannasi, "A New Digital Signature Algorithm for Ensuring the Data Integrity in Cloud using Elliptic Curves," *Int. Arab J. Inf. Technol.*, vol. 18, no. 2, pp. 180–190, 2021, doi: 10.34028/iajit/18/2/6.

[17] . Tan Soofun, Nur Sufia Binti Ahmad Zulkifli, Florence Sia, "Enhancing Electronic Document Security with Lightweight Digital Signature," *IEEE Int. Conf. Artif. Intell. Eng. Technol.*, 2024, doi: 10.1109/IICAIET62352.2024.10730678.

[18] . M. A. Gani, E. Budhiarti Nababan and A. Candra, "Enhancing Digital Document Security with Elliptic Curve Digital Signature Algorithm and Watermarking Techniques," *Int. Conf. Creat. Commun. Innov. Technol.*, pp. 1–7, 2024, doi: 10.1109/ICCIT62134.2024.10701112.

[19] J. F. Sinan Salih,Ravi Sekhar, "Integrated Digital Signature Based Watermarking Technology for Securing Online Electronic Documents," *Fusion Pract. Appl.*, vol. 14, no. 1, pp. 120–128, 2024, doi: https//doi.org/10.54216/FPA.140111.

[20] M. K. Kabier, A. A. Yassin, Z. A. Abduljabbar, and H. S. Hashim, "Using a QR-Code System and a Digital Signature to Authenticate Paper-Based Documents in Schools in Iraq," *Proc. - 9th Int. Conf. Comput. Softw. Model. ICCSM 2025*, pp. 159–163, 2025, doi: 10.1109/ICCSM66818.2025.00035.

[21] . Wisnu Uriawan,Ray Ramadita, "Authenticate and Verification Source Files using

SHA256 and HMAC Algorithms," *Reseachgate*, 2024, doi: 10.20944/preprints202407.0075.v1.

[22] Venkatesh B, L Swathi, "Implementation and Evaluation of Data Protection in Databases Using Symmetric Encryption Algorithms," *J. Neonatal Surg.*, vol. 14, no. 8, 2025, [Online]. Available: https://www.jneonatalsurg.com/index.php/jns/article/view/2558

[23] Ziyu Jiang,Hongxia Wang,SongYuanHun, "A robust PDF watermarking scheme with versatility and compatibility," *Multimed Tools Appl*, vol. 83, pp. 64341–64367, 2024, doi: https://doi.org/10.1007/s11042-024-18151-w.

[24] Sadaf Hussain, Rabia Afzaal, "Lightweight Message Authentication Protocol for Low-Resource IoT Devices," *Int. J. Emerg. Eng. Technol.*, vol. 2, no. 1, 2023, [Online]. Available: https://pjosr.com/index.php/ijeet/article/view/932

[25] . T.S. Putra,M.A Budiman,S.Suwilo, "Signcryption Techniques For Digital File Security Using the RSA Multi-Factor Algorithm and the ESIGN Algorithm," *Int. Conf. Comput. Sci. Inf. Technol.*, pp. 1–6, 2023, doi: 10.1109/ICoSNIKOM60230.2023.10364520.

[26] K.Somsuk, "The development of signing and verification methods for high speed digital signatures on electronic official documents by using RSA cryptography," *Cogent Eng.*, 2024, doi: https://doi.org/10.1080/23311916.2024.2432513.

[27] Suhardi Suhardi, "Use of QRCode and Digital Signature Using The DSA Method to Authenticate Student Academic Documents," *J. Comput. Networks*, vol. 6, no. 4, 2024, doi: 10.47709/cnahpc.v6i4.4765.

[28] Theophilus Wellem, Yessica Nataliani, "Academic Document Authentication using Elliptic Curve Digital Signature Algorithm and QR Code," *JOIV Int. J. Informatics Vis.*, vol. 6, no. 3, p. 667, 2022, doi: 10.30630/joiv.6.2.872.

[29] M. Bisheh-Niasar, R. Azarderakhsh and M. Mozaffari-Kermani, "Cryptographic Accelerators for Digital Signature Based on Ed25519," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 29, no. 7, pp. 1297–1305, 2021, doi: 10.1109/TVLSI.2021.3077885.

[30] Mike Yuliana, Wildan Dharma Walidaniy, "Efficient Multi-signature and QR Code Integration for Document Authentication Using EdDSA-based Algorithm," *Int. J. Intell. Eng. Syst.*, vol. 17, no. 2, 2024, doi: 10.22266/ijies2024.0430.32.

[31] K. S. Velrani, N. Lokeshvaran, and D. Jerome Richard, "Integrating Hybrid Cryptography with Role-Based Access Management in Multi-Tier Cloud File Encryption," *Proc. ICECONF 2025 - 2025 2nd Int. Conf. Artif. Intell. Knowl. Discov. Concurr. Eng.*, 2025, doi: 10.1109/ICECONF65644.2025.11379450.